What is claimed is:

1. A method for counting performance events for a computing system, wherein the system includes i) a processor having a set of on-chip, performance monitoring counter registers and ii) system memory, the method comprising the steps of:

a) designating a first one of the counters as a low-order counter for counting a certain performance event encountered by the processor;

b) associating with the first counter a second one of the counters as a high-order counter for the performance event;

c) incrementing the first counter responsive to detecting the performance event for a first processing thread;

d) updating, responsive to a second thread becoming active, an accumulator in system memory for the first thread and first and second counters;

e) loading, responsive to the first thread becoming active, values of the first and second counters from the accumulator, so that while the first thread is active the values of the first and second counters provide a consistent meaning relative to values that were read during a previous time when the first thread was active, despite any intervening thread switches; and

f) performing, responsive to a user call to read and return a combined value from the first and second counters, an operation comprising the steps of:

reading a first instance of the second counter;

reading the first counter; and

reading a second instance of the second counter before returning the combined value.

2. The method of claim 1, wherein the updating handles certain bits of the accumulator as most-significant bits (MSB's), certain other bits of the accumulator as least-significant bits (LSB's) and certain bits of the first counter as LSB's, and the updating includes:

comparing the LSB's of the first counter to the LSB's of the accumulator;

incrementing the MSB's of the accumulator if the LSB's of the first counter are a smaller value than the LSB's of the accumulator; and

overwriting the LSB's of the accumulator with the LSB's of the first counter.

3. The method of claim 1, wherein the updating handles certain bits of the accumulator as most-significant bits (MSB's), certain other bits of the accumulator as least-significant bits (LSB's), certain bits of the first counter as LSB's and certain other bits of the first counter are handled as overlapping bits and the updating includes:

adding the overlapping bits of the first counter to the MSB's of the accumulator; and

overwriting the LSB's of the accumulator with the LSB's of the first counter.

4. The method of claim 1, wherein the loading handles certain bits of the second counter as MSB's and the loading includes:

overwriting the MSB's of the second counter with the MSB's of the accumulator; and

overwriting the LSB's of the first counter with the LSB's of the accumulator.

5. The method of claim 4, wherein certain other bits of the first counter are handled as overlapping bits and the loading includes:

resetting the overlapping bits of the first counter.

6. The method of claim 1, wherein the operation of step f) comprises the steps of:

comparing values of the second counter read in the first and second instances; and

merging bit values of the first and second counters into a combined value if the second counter are the same in the first and second instances.

7. The method of claim 6, wherein if the second counter is different in the first and second instances the operation of step f) comprises the steps of:

reading additional instances of the first and second counters;

comparing and repeating the reading of additional instances of the first and second counters until values of the second counter read in the last two instances are the same;

and

thereafter merging bit values of the first and second counters into a combined value.

8. An apparatus for counting performance events for a computing system, wherein the system includes i) a processor having a set of on-chip performance monitoring counter registers and ii) system memory, wherein the apparatus includes instructions with which the processor is operable to perform the steps of:

a) designating a first one of the counters as a low-order counter for counting a certain performance event encountered by the processor;

b) associating with the first counter a second one of the counters as a high-order counter for the performance event;

c) incrementing the first counter responsive to detecting the performance event for a first processing thread;

d) updating, responsive to a second thread becoming active, an accumulator in system memory for the first thread and first and second counters;

e) loading, responsive to the first thread becoming active, values of the first and second counters from the accumulator, so that while the first thread is active the values of the first and second counters provide a consistent meaning relative to values that were read during a previous time when the first thread was active, despite any intervening thread switches; and

f) performing, responsive to a user call to read and return a combined value from the first and second counters, an operation comprising the steps of:

reading a first instance of the second counter;

reading the first counter; and

reading a second instance of the second counter before returning the combined value.

9. The apparatus of claim 8, wherein the updating handles certain bits of the accumulator as most-significant bits (MSB's), certain other bits of the accumulator as least-significant bits (LSB's) and certain bits of the first counter as LSB's and wherein the apparatus includes instructions with which the processor is operable to perform steps in which the updating includes:

comparing the LSB's of the first counter to the LSB's of the accumulator;

incrementing the MSB's of the accumulator if the LSB's of the first counter are a smaller value than the LSB's of the accumulator; and

overwriting the LSB's of the accumulator with the LSB's of the first counter.

10. The apparatus of claim 8, wherein the updating handles certain bits of the accumulator as most-significant bits (MSB's), certain other bits of the accumulator as least-significant bits (LSB's), certain bits of the first counter as LSB's and certain other bits of the first counter are handled as overlapping bits and wherein the apparatus includes instructions with which the processor is operable to perform steps in which the updating includes:

adding the overlapping bits of the first counter to the MSB's of the accumulator; and

overwriting the LSB's of the accumulator with the LSB's of the first counter.

11. The apparatus of claim 8, wherein the loading handles certain bits of the second counter as MSB's and wherein the apparatus includes instructions with which the processor is operable to perform steps in which the loading includes:

overwriting the MSB's of the second counter with the MSB's of the accumulator; and

overwriting the LSB's of the first counter with the LSB's of the accumulator.

12. The apparatus of claim 11, wherein the loading handles certain other bits of the first counter as overlapping bits and wherein the apparatus includes instructions with which the processor is operable to perform steps in which the loading includes:

resetting the overlapping bits of the first counter.

13. The apparatus of claim 8, wherein the apparatus includes instructions with which the processor is operable to perform steps in which the operation of step f) includes:

comparing values of the second counter read in the first and second instances; and

merging bit values of the first and second counters into a combined value if the second counter are the same in the first and second instances.

14. The apparatus of claim 13, wherein the apparatus includes instructions with which the processor is operable to perform steps in which the operation of step f) includes the following steps if the second counter is different in the first and second instances:

reading additional instances of the first and second counters;

comparing and repeating the reading of additional instances of the first and second counters until values of the second counter read in the last two instances are the same;

and

thereafter merging bit values of the first and second counters into a combined value.

15. A computer program product for counting performance events for a computing system, wherein the system includes i) a processor having a set of on-chip performance monitoring counter registers and ii) system memory, the computer program product comprising:

instructions for designating a first one of the counters as a low-order counter for counting a certain performance event encountered by the processor;

instructions for associating with the first counter a second one of the counters as a high-order counter for the performance event;

instructions for incrementing the first counter responsive to detecting the performance event for a first processing thread;

instructions for updating, responsive to a second thread becoming active, an accumulator in system memory for the first thread and first and second counters;

instructions for loading, responsive to the first thread becoming active, values of the first and second counters from the accumulator, so that while the first thread is active the values of the first and second counters provide a consistent meaning relative to values that were read during a previous time when the first thread was active, despite any intervening thread switches; and

instructions for performing, responsive to a user call to read and return a combined value from the first and second counters, a certain operation comprising the steps of:

reading a first instance of the second counter;

reading the first counter; and

reading a second instance of the second counter before returning the combined value.

16. The computer program product of claim 15, wherein the updating handles certain bits of the accumulator as most-significant bits (MSB's), certain other bits of the accumulator as least-significant bits (LSB's) and certain bits of the first counter as LSB's and wherein the instructions for updating include:

instructions for comparing the LSB's of the first counter to the LSB's of the accumulator;

instructions for incrementing the MSB's of the accumulator if the LSB's of the first counter are a smaller value than the LSB's of the accumulator; and

instructions for overwriting the LSB's of the accumulator with the LSB's of the first counter.

17. The computer program product of claim 15, wherein the updating handles certain bits of the accumulator as most-significant bits (MSB's), certain other bits of the accumulator as least-significant bits (LSB's), certain bits of the first counter as LSB's and certain other bits of the first counter are handled as overlapping bits and wherein the instructions for updating include:

instructions for adding the overlapping bits of the first counter to the MSB's of the accumulator; and

instructions for overwriting the LSB's of the accumulator with the LSB's of the first counter.

18. The computer program product of claim 15, wherein the loading handles certain bits of the second counter as MSB's and the instuctions for loading include

overwriting the MSB's of the second counter with the MSB's of the accumulator; and

overwriting the LSB's of the first counter with the LSB's of the accumulator.

19. The computer program product of claim 18, wherein certain other bits of the first counter are handled as overlapping bits and the instructions for loading include:

instructions for resetting the overlapping bits of the first counter.

20. The computer program product of claim 19, wherein the certain operation includes the steps of:

comparing values of the second counter read in the first and second instances; and

merging bit values of the first and second counters into a combined value if the second counter are the same in the first and second instances.

21. The computer program product of claim 20, wherein the certain operation includes the steps of:

reading additional instances of the first and second counters;

comparing and repeating the reading of additional instances of the first and second counters until values of the second counter read in the last two instances are the same;

and

thereafter merging bit values of the first and second counters into a combined value.